SPECIFICATION

Electronic Version 1.2.8
Stylesheet Version 1.0

QUEUE SCHEDULING MECHANISM IN A DATA PACKET TRANSMISSION SYSTEM

Background of the Invention

[0001] FIELD OF THE INVENTION

[0002] The present invention relates to a data packet transmission system wherein the data packets are transmitted from an input device to an output device through a switch engine. In particular, the present invention relates to a queue scheduling mechanism in such a data packet transmission system.

[0003] BACKGROUND OF THE INVENTION

[0004] In today's world of telecommunications, characterized by an insatiable demand for bandwidth, there are two very fast growing technology sectors. These two technology sectors are the Internet and wireless communications. The Internet is primarily concerned with moving data while wireless communications is still mainly dealing with voice transmission. However, all of this is changing very rapidly. Service providers of all types tend to offer more services in an attempt to become, or to remain, profitable. Service offerings range from long distance transport of voice and data over high-speed data backbone to the Internet and data services being offered on wireless pieces of equipment especially wireless phones of second and third generations.

[0005] Voice has long been transported in the form of data on circuit-switched Time Division Multiplexed (TDM) networks which are very different from the Internet packet networks obeying the Internet Protocol (IP). TDM is a connection oriented network while IP is connectionless. Hence, TDM can offer the carrier-grade type of service required by delay-sensitive applications, such as voice, while IP is well adapted to the

transport of data.

[0006] All specialized transport network operators want to converge to a similar "one-fits-all" type of network, i.e. a packet network able to process different flows of data depending on Quality of Service (QoS) schemes so that flows are indeed processed according to some specific requirements such as delay, jitter, bandwidth, and packet loss.

[0007] Switching and routing have been opposed due to the manner in which data packets flow through the nodes of the network. Switching is tightly associated to connection oriented protocols like ATM and requires that a path be established prior to any data movement while routing is essentially the mode of operation of IP, and its hop-by-hop moving of data packets, with a decision to be made at each node. However, the end result is that whichever access protocol is in use, the networks are in actuality becoming switched-packet networks.

[0008] When packets arrive in a node, the layer 2 forwarding component of the switching node searches a forwarding table to make a routing decision for each packet. Specifically, the forwarding component examines information contained in the packet's header, searches the forwarding table for a match, and directs the packet from the input interface to the output interface across the switch engine.

[0009] Generally, a switching node includes a plurality of output queues corresponding respectively to the plurality of output adapters and a shared memory for temporarily storing the incoming packets to be switched. The switch architecture is known to potentially provide the best possible performance allowing a full outgoing throughput utilization with no internal blocking and minimum delay.

[0010] Every queue is also organized by priority. That is, incoming packet headers, which carry a priority tag, are inspected not only to temporarily store packets in different queues, according to the output ports they are due to leave the switch engine but also are sorted by priority within each queue so that higher priority packets are guaranteed to be admitted first in the shared memory, getting precedence over lower priority traffic. In turn, the switch engine applies the same rule to the admitted packets, always privileging higher priorities. This is achieved by organizing the output queues

by priority too. Hence, packet pointers, in each output queues are sorted so that admitted packets of higher priorities exit the switch engine first even though older packets, yet of a lower priority, are still waiting.

[0011] Generally, the priorities associated with the data packets are fully pre-emptive. Thus, if there are four priorities from P_0 to P_3, priority P_0 is going to take immediate precedence over any other traffic at priorities P_1-P_3 and so on. This is definitely a feature necessary to be able to handle a mix of voice and real-time traffic along with "pure" data traffic over a single network. This guarantees that data for the "pure" data traffic type of applications are handled with no delay so that there is no latency other than the necessary minimum time to traverse the switch engine and, even more importantly, in order that no significant jitter be added to any flow of real-time packets. However, this is necessarily done at the expense of lower priority traffic which has, in case of congestion, to wait. Even if this is not a problem since the transfer of data files is normally insensitive to delay and jitter, a lower priority (e.g. P_3) may be completely starved by higher priorities (e.g. P_0-P_2).

Brief Summary of the Invention

[0012] Accordingly, an object of the present invention is to provide a queue scheduling mechanism which avoids a lower priority being starved by higher priorities except for one or more higher priorities considered as "exhaustive priorities" which may never be preempted by lower priorities.

[0013] Another object of the present invention is to provide a queue scheduling mechanism including both a credit device that enables a minimum bandwidth to the lower priority traffic and an exhaustive priority register that registers one or several priorities which may never be preempted.

[0014]

The present invention relates therefore to a queue scheduling mechanism in a data packet transmission system, the data packet transmission system including a transmission device for transmitting data packets, a reception device for receiving the data packets, a set of queue devices respectively associated with a set of priorities each defined by a priority rank for storing each data packet transmitted by the transmission device into the queue device corresponding to its priority rank, a queue

scheduler for reading, at each packet cycle, a packet in one of the queue devices, and a credit device that provides, at each packet cycle, a value N defining the priority rank to be read by the queue scheduler from the queue device corresponding to the priority N instead of the queue device determined by a normal priority preemption algorithm. The queue scheduling mechanism further includes an exhaustive priority register that registers the value of at least one exhaustive priority rank to be read by the queue scheduler from the queue device corresponding to the exhaustive priority rank rather than from the queue device corresponding to the priority N.

Brief Description of the Several Views of the Drawings

[0015] The above and other objects, features and advantages of the present invention will be better understood by reading the following more particular description of the invention in conjunction with the accompanying drawings wherein:

[0016] FIG. 1 illustrates a block-diagram representing schematically a switch device wherein a queue scheduling mechanism according to the present invention is implemented; and,

[0017] FIGS. 2A and 2B together illustrate a flow chart showing the steps of the method implemented in the queue scheduling mechanism according to the present invention.

Detailed Description of the Invention

[0018] A queue scheduling mechanism disclosed in commonly assigned European Patent Application No. 01480118.7, herein incorporated by reference, includes a credit table that provides at each packet cycle a value N defining the priority rank to be considered by the queue scheduler, whereby a data packet is read by the queue scheduler from the queue device corresponding to the priority N instead of the queue device determined by the normal priority preemption algorithm. However, in some configurations, it is required by the customer that one or several high priorities never be preempted by lower priorities such as in the case for a communication link transmitting essentially voice or video data.

[0019] The queue scheduling mechanism according to the present invention is, in a preferred embodiment, implemented in a switch engine of a switching node wherein data packets are received from a plurality of input adapters and sent through the

switch engine to another plurality of output adapters. However, such a queue scheduling mechanism could be used in any system wherein data packets received from transmitting devices are stored in queues according to several priorities before being read under the control of a queue scheduling mechanism for being sent to receiving devices.

[0020] Referring to FIG. 1, a switch engine 10 wherein the present invention is implemented, comprises several queue devices 12, 14, 16 and 18 generally organized as First-In-First-Outs (FIFOs) respectively associated with priority ranks P_0, P_1, P_2 and P_3. This means that data packets having a priority P_0 are stored in queue device 12, data packets of priority P_1 in queue device 14, data packets in priority P_2 in queue device 16, and data packets of priority P_3 in queue device 18.

[0021] At each packet cycle, the queue devices 12, 14, 16 and 18 have to be scheduled by a queue scheduler 20 through control lines 21 to allow a data packet to be read and sent to an output adapter 22 wherein the packet is stored in a queue device 24. However, a data packet may be read from a queue device of the switch engine 10 only if a GRANT signal sent on line 26 from the queue device 24 to the queue scheduler 20 is active. The activation of the GRANT signal for a given priority depends upon an algorithm which is a function of the filling level of queue device 24. Generally, there are several filling thresholds associated respectively with the priority ranks which make the GRANT signal inactive for a priority rank when the threshold associated with this priority rank is reached. Note that a packet of a priority N is read from the corresponding queue device 12, 14, 16 or 18 only if there is at least one packet stored in this queue device. The queue scheduler 20 is aware of this by means of lines 25 from the queue devices.

[0022] In order to avoid having a data packet with a low priority from staying in the switch engine 10 for a very long time due to highest priority traffic resulting in holding a switch resource which prevents lowest priority packets from being queued and setting a time out at the end user level followed by a retransmission of the low priority data packet which increases network congestion, the switch engine 10 is also provided with a credit table 28 which enables to guarantee a minimum bandwidth for any priority rank. The credit table 28, which is programmable, indicates which priority

is allowed to be served unconditionally at each packet cycle, thus overriding the normal preemptive priority mechanism. Such a credit table 28 can be a RAM memory having 256 locations wherein the address to be read is incremented at each packet cycle, the address returning to 0 when it reaches the value 255. For example, the credit table 28 can be organized as shown in Table 1:

[0023]

[t1]

Address	Priority
0	P3
1	P2
2	P1
:	:
12	P1
:	:
21	P2
22	P1
:	:
32	P1
:	:
41	P2
42	P1
:	:
100	P3
:	:

[0024] The number of locations containing each value N is defined according to a predetermined percentage of occurrences with respect to the values of the other priority ranks. In the present case, it can be seen that the priority P_3 is registered at addresses 0, 100..., that is, P_3 is in one location out of one hundred locations of the credit table 28; the priority P_2 is registered at addresses 1, 21, 41, ..., that is, P_2 is in one location out of twenty locations of the credit table 28; and, the priority P_1 is registered at addresses 2, 12, 22 ..., that is, P_1 is in one location out of ten locations of the credit table 28. The other address locations not having a priority registered means the priority P_0 is registered by default since, in such a case, it is the priority P_0 which is served first before the other priorities.

[0025] Accordingly, the credit provided to the different priority ranks is the following in percentage :

[0026] P_0 84%

[0027] P_1 10%

[0028] P_2 5%

[0029] P_3 1%

[0030] In order to avoid that a lower priority given by the credit table 28 may preempt the highest priority (P_0),or several of the highest priorities, the switch engine 10 is provided with an exhaustive priority register 30 that registers, or records, the exhaustive priorities. At each packet cycle, the exhaustive priority register 30 is read by the queue scheduler 20 to determine whether there is a data packet having an exhaustive priority which is waiting to be transmitted. It is only when there is no such data packet that a data packet of the priority rank pointed in credit table 28 may be transmitted.

[0031] The method for implementing the queue scheduling mechanism according to the present invention is illustrated by the flow chart of FIGS. 2A and 2B. At each packet cycle, a variable n is set to 0 (step 40). It is then checked whether the GRANT signal is active for priority n (an exhaustive priority) or, in other words, whether there is authorization to send a packet having the priority 0 since n = 0 (step 42). If so, it is determined whether there is a priority 0 packet to be read in the queue corresponding to priority 0 (step 44). If it is the case, a priority 0 packet is read in the corresponding queue and sent to the output device (step 46). Then, the address of the credit table is incremented (step 48) and the process is looped back to step 40.

[0032] If the signal GRANT is not active for the priority 0, or if there is no priority 0 packet in the corresponding queue, it is determined whether there are other exhaustive priorities further to the priority 0 such as priority 1, 2 ... (step 50). If so, the variable n is incremented to n+1, i.e. from 0 to 1 in the present example (step 52) and the above processing is repeated in order to send a priority 1 packet. Such a processing is repeated until there is no more exhaustive priority.

[0033] Then the credit table is read (step 54) to know the priority rank which is recorded at the address being read at this cycle. It is assumed that the priority rank being recorded is the priority N, N being a number different from 0 as mentioned above or 0 by default. It is then checked whether the GRANT signal is ON for this priority, that is whether there is authorization to send a priority N packet (step 56). If so, it is determined whether there is a packet to be read in the queue corresponding to priority N (step 58). If it is the case, a priority N packet is read in the corresponding queue and sent to the output device (step 60). Then, the address of the credit table 28 is incremented (step 48) and the process is looped back to step 40.

[0034] If the signal GRANT is not active for the priority N which has been read from the credit table 28 or if there is no priority N packet in the corresponding queue, it is then checked whether there is authorization to send a priority n+1 packet (the GRANT signal is active) for the considered priority (step 62), that is the highest priority after the exhaustive priorities. If so, it is determined whether there is a packet to be read in the queue corresponding to the priority n+1 (step 64). If it is the case, a priority n+1 packet is read from the queue corresponding to this priority and sent to the output device (step 66). Then, the address of the credit table 28 is incremented (step 48) and the process is looped back to step 40.

[0035] If the signal GRANT is not active for the priority n+1 or if there is no priority n+1 packet in the corresponding queue, it is checked whether the value of n+1 has reached the value M corresponding to the lowest priority (step 68). If so, the address of the credit table 28 is incremented and the process is looped back to step 40. If it is not the case, variable n is incremented to n+1 (step 70) and the process returns to step 62 of processing the packet of priority n+1, and so on.

[0036] It must be noted that, if there are a credit table and an exhaustive priority register in the switch engine as described in reference to FIG. 1 and not in the input adapter and the output adapter, there is a risk that the lower priority data packets may not be scheduled and stay in the adapter queue as long as there is higher priority traffic. It is therefore necessary that a credit table with the same percentage of the priority ranks (e.g. 1% for P3, 5% for P2 and 10% for P1 as seen above) and an exhaustive priority register recording the same exhaustive priorities exist in the input adapter as well as

in the output adapter.

[0037] Although specific embodiments of the present invention have been illustrated in the accompanying drawings and described in the foregoing detailed description, it will be understood that the invention is not limited to the particular embodiments described herein, but is capable of numerous rearrangements, modifications and substitutions without departing from the scope of the invention. The following claims are intended to encompass all such modifications.